ООО «АЛЬФА-ИНТЕГРАТОР-ИНФОЭНЕРГО»

Автоматизированная информационная система управления процессами капитального строительства (КапС.Pro)

ИНСТРУКЦИЯ АДМИНИСТРАТОРА
(ДЛЯ УСТАНОВКИ ПО)

АИС Капс.Pro

Санкт-Петербург
2021

Оглавление

Принятые термины и сокращения

Термин	Определение
АИС КапС.Pro	Автоматизированная информационная система управления процессами капитального строительства (КапС.Pro)
ПО	Программное обеспечение
СУБД	Система управления базами данных

1. Общие положения

Настоящая Инструкция системного администратора разработана с целью:

- описания архитектуры системы и взаимосвязи её компонентов;
- описания процесса установки и настройки системы;
- определения порядка диагностирования проблем функционирования системы;
- упорядочения работы должностных лиц, связанной с диагностированием проблем функционирования системы.

В настоящем документе регламентируются действия при выполнении следующих мероприятий:

- установка и настройка системы;
- диагностирование проблем функционирования системы.

2. Описание архитектуры АИС КапС.Pro

АИС КапС.Pro представляет собой программно-аппаратный комплекс, состоящий из серверов приложений, сервера базы данных и установленного на них ПО.

Схема развертывания АИС КапС.Pro в инфраструктуре Заказчика представлена на Рис.1

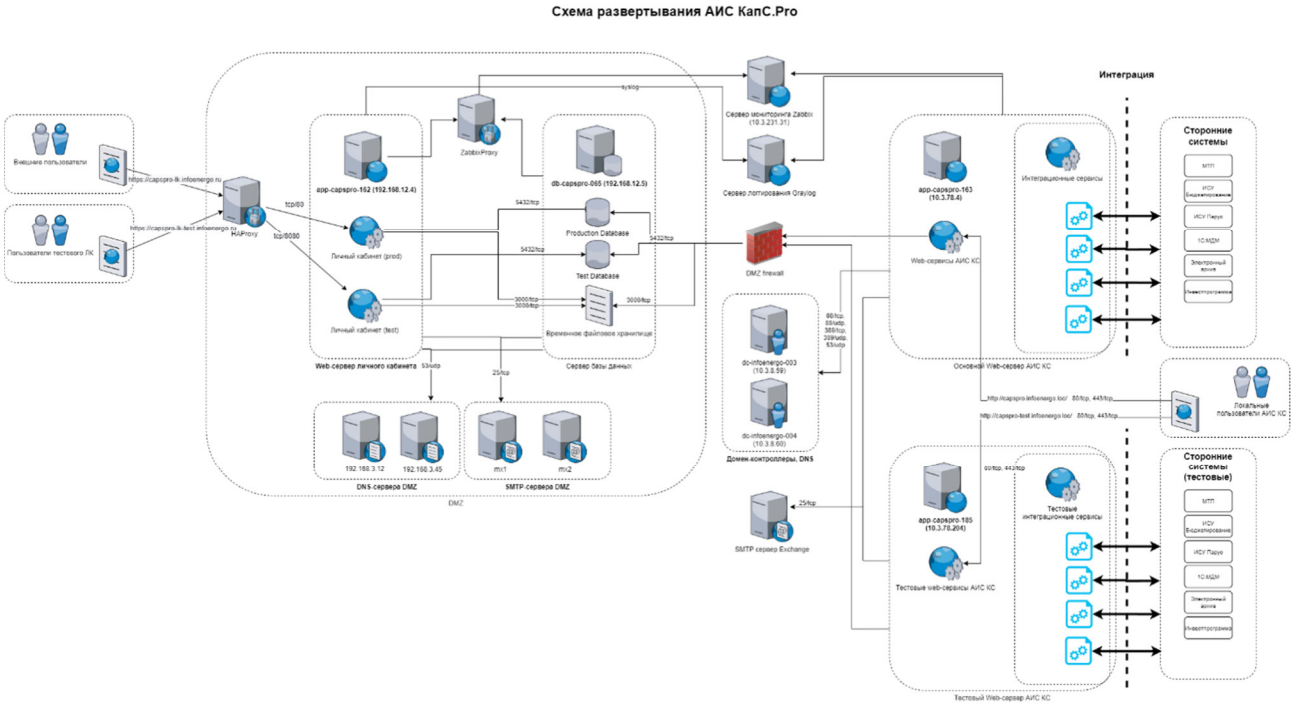

Рис 1. Схема развертывания АИС КапС.Pro

АИС КапС.Pro включает в себя следующие серверы:

app-capspro-163 (10.3.78.4) - Основной сервер АИС КапС.Pro

app-capspro-162 (192.168.12.4) –Сервер личного кабинета;

db-capspro-065 (192.168.12.5) - Сервер баз данных;

app-capspro-185 (10.3.78.204) - Тестовый сервер АИС КапС.Pro.

На всех серверах операционной системой является РЕД ОС версии 7.3

Сервер приложений (также установлен на всех серверах) Node.JS версии 18.12

Сервер приложений (также установлен на всех серверах) ASP.NET Core Runtime версии 3.1

На сервере базы данных установлена СУБД Postgres Pro Standard 14.5

На основном сервере АИС КапС.Pro также установлено следующее ПО:

- krb5-libs krb5-workstation - библиотеки для обеспечения SSO-авторизации пользователей по протоколу Kerberos
- libgdiplus - графическая библиотека для обеспечения работы сервиса печати отчетов FastReport

Схема взаимодействия АИС КапС.Pro со сторонними системами показана на Рис 2:

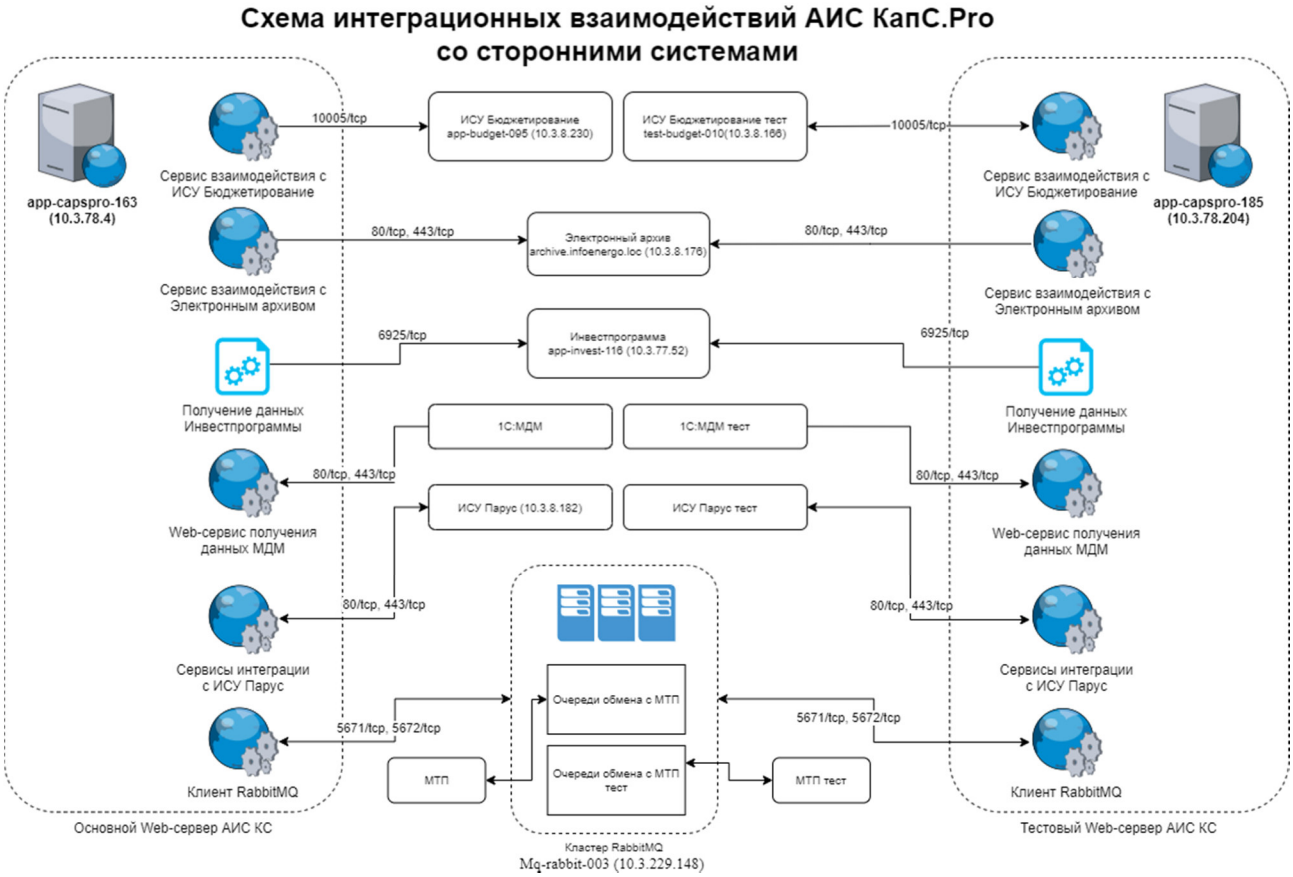

Рис.2. Схема интеграционных взаимодействий АИС КапС.Pro

3. Установка и настройка системы

3.1 Сервисы под управлением Node.JS
3.1.1 Установка и настройка Node.JS
Установка сервера приложений производится на всех серверах АИС КапС.Pro

3.1.1.1 Проверить доступность модуля Node.JS соответствующей версии в репозитории и включить нужный поток в случае необходимости:
sudo dnf module list nodejs
sudo dnf module enable nodejs:18

3.1.1.2 Установить модуль Node.JS версии 12 из репозитория:
sudo dnf module install nodejs:18

3.1.1.3 Проверить установленную версию Node.JS и npm:
node -v
npm –v

3.1.1.4 В случае, если необходимо будет запускать web-приложения на портах <1024, предоставить привилегии на привязку к этим портам:
sudo setcap 'cap_net_bind_service=+ep' $(readlink -f $(which node))

3.1.1.5 Открыть необходимые порты для подключения к приложениям на firewall-е (в данном случае приложение на порту 3000/tcp):

- добавляем порт
  sudo firewall-cmd --permanent --zone=public --add-port=3000/tcp
- перезапускаем firewall
  sudo firewall-cmd --reload
- проверяем включение правила
  sudo firewall-cmd --list-all

3.1.1.6 Создать и задать пароль пользователю webuser, от имени которого будут выполняться web-приложения:
useradd webuser
passwd webuser

3.1.1.7 В случае, если доступ сервера в интернет осуществляется через прокси-сервер, установить параметры доступа для npm через него:
npm config set https-proxy http://x.x.x.x:xxxx
npm config set proxy http:// x.x.x.x:xxxx

3.1.1.8 Установить менеджер процессов pm2
sudo npm install -g pm2

3.1.1.9 В случае, если доступ сервера в интернет осуществляется через прокси-сервер, установить параметры доступа для npm от имени пользователя webuser через него:
sudo su - webuser
npm config set https-proxy http://x.x.x.x:xxxx
npm config set proxy http:// x.x.x.x:xxxx

3.1.2 Запуск и мониторинг сервисов
Для поддержания непрерывной работы сервисов под управлением Node.JS создается файл ecosystem.config.js в каталоге проекта, например:

```
'use strict';
module.exports = {
  apps: [{
    name: 'aisks',
    script: 'server/index.js',
    watch: true,
    exp_backoff_restart_delay: 100,
    env: {
      'CAPCON_PORT': 80,
      'CAPCON_SSL_PORT': 443,
      'NODE_ENV': 'aisks-dev',
      'CAPCON_LOG_DIR': '/home/aisks/logs/capcon',
      'CAPCON_FILE_ROOT': '/home/aisks/data/files',
      'CAPCON_ESTIMATES_ROOT': '/home/aisks/data/estimate',
      'CAPCON_SSL_CERTIFICATE': '/home/aisks/ssl/aisks.pem',
```

```
    'CAPCON_PRIVATE_KEY': '/home/aisks/ssl/aisks.key'
   },
   env_production: {
    'CAPCON_PORT': 80,
    'CAPCON_SSL_PORT': 443,
    'NODE_ENV': 'production',
    'CAPCON_LOG_DIR': '/home/webuser/logs/capcon',
    'CAPCON_FILE_ROOT': '/home/webuser/data/files',
    'CAPCON_SSL_CERTIFICATE': '/home/webuser/ssl/aisks.pem',
    'CAPCON_PRIVATE_KEY': '/home/webuser/ssl/aisks.key'
   }
  }]
};
```

Для запуска и мониторинга приложений используется менеджер процессов pm2.
pm2 start ecosystem.config.js --only aisks --env production # создает приложение aisks, используя переменные среды из окружения production на основе файла ecosystem.config.js
pm2 restart aisks # перезапускает приложение aisks
pm2 stop aisks # останавливает приложение aisks
pm2 logs 2 --lines 200 # отображает последние 200 строк лога приложения с id=2

3.2 Установка и настройка Postgres Pro
Установка СУБД Postgres Pro производится на сервере баз данных.
3.2.1 Проверить доступность модуля Postgres Pro соответствующей версии в репозитории и включить необходимый поток в случае необходимости:
sudo dnf module list Postgres Pro
sudo dnf module enable Postgres Pro:10

3.2.2 Установить Postgres Pro версии 10 из репозитория:
sudo dnf install Postgres Pro-server Postgres Pro-contrib

3.2.3 Открыть порт для подключения к Postgres Pro на firewall-е (по-умолчанию это порт 5432/tcp):

- добавляем порт
  sudo firewall-cmd --permanent --zone=public --add-port=5432/tcp
- перезапускаем firewall
  sudo firewall-cmd --reload
- Проверяем включение правила
  sudo firewall-cmd --list-all

3.2.4 Инициализировать базу данных
sudo Postgres Pro-setup initdb

3.2.5 Запустить и включить в автозапуск службу
sudo systemctl enable --now Postgres Pro

3.2.6 Проверить доступность и установленную версию СУБД:
sudo -u postgres psql -c "SELECT version();"

3.2.7 Задать пароль пользователю postgres, от имени которого запускается СУБД:

sudo passwd postgres

3.2.8 Задать пароль суперпользователю СУБД postgres:
sudo -u postgres psql
\password postgres
\q

3.2.9 Настроить парольную аутентификацию в файле конфигурации pg_hba.conf (расположение по-умолчанию $HOME/data/pg_hba.conf в каталоге пользователя postgres):

```
# "local" is for Unix domain socket connections only
local   all         all                               md5
# IPv4 local connections:
host    all         all         127.0.0.1/32          md5
# IPv6 local connections:
host    all         all         ::1/128               md5
# LAN connections
host    all         all         0.0.0.0/0             md5
```

В разделе LAN connections можно ограничить доступ только определенными сетями, указав соответствующую маску

3.2.10 Настроить следующие параметры в файле конфигурации Postgres Pro.conf (расположение по-умолчанию $HOME/data/Postgres Pro.conf в каталоге пользователя postgres):

```
#Включаем прослушивание на внешних интерфейсах системы
listen_addresses = '*'
#Устанавливаем количество соединений
max_connections = 100
#Включаем уровень записи в WAL, необходимый для восстанвления из резервной копии
wal_level = replica
#Включаем режим архивации
archive_mode = on
#Устанавливаем команду архивации, вместо /opt/Postgres Pro/wal_backup/ должен быть каталог, в котором будут размещаться резервные копии WAL
archive_command = 'test ! -f /opt/Postgres Pro/wal_backup/%f.gz && /usr/bin/gzip -c %p > /opt/Postgres Pro/wal_backup/%f.gz'
#Настраиваем полтику удержания WAL
wal_keep_segments = 60
#Параметры локализации
lc_messages = 'en_US.UTF-8'
lc_monetary = 'ru_RU.UTF-8'
lc_numeric = 'ru_RU.UTF-8'
lc_time = 'ru_RU.UTF-8'
#Параметры отображения даты
datestyle = 'iso, mdy'
```

3.2.11 Перезапустить сервис Postgres Pro для применения новых параметров:
sudo systemctl restart postgresql

3.2.12 Настройка резервного копирования
Создать скрипт резервного копирования. В данном примере каталог запуска Postgres Pro - /opt/postgresql, на сервере хранятся 2 последних базовых резервных копии (в каталоге db_backup) и WAL за последние 2-е суток (в каталоге wal_backup):

```
#!/bin/bash
```

```
PG_HOME=/opt/ postgresql
export PG_HOME
mkdir $PG_HOME/pg_backup
/usr/bin/pg_basebackup -U postgres -D $PG_HOME/pg_backup -Ft -z -Xf

INDEX=$(date +"%u")

test -e $PG_HOME/db_backup/base.${INDEX}.tar.gz && rm $PG_HOME/db_backup/base.${INDEX}.tar.gz
cp $PG_HOME/pg_backup/base.tar.gz $PG_HOME/db_backup/base.${INDEX}.tar.gz

test -e $PG_HOME/db_backup/base.last.tar.gz && rm $PG_HOME/db_backup/base.last.tar.gz
ln $PG_HOME/db_backup/base.${INDEX}.tar.gz $PG_HOME/db_backup/base.last.tar.gz

rm -r $PG_HOME/pg_backup
test -e $PG_HOME/db_backup/base.${INDEX}.tar.gz && test -e $PG_HOME/db_backup/base.$(date --date="-1 day" +"%u").tar.gz && find $PG_HOME/db_backup -type f -mtime +1 -exec rm {} \;
find $PG_HOME/wal_backup -type f -mtime +2 -exec rm {} \;
```

3.3 Сервисы под управлением .NET Core

3.3.1 Установка среды выполнения .NET Core

Установка сервера приложений производится на всех серверах АИС КапС.Pro.

sudo dnf install aspnetcore-runtime-3.1

3.3.2 Запуск и мониторинг сервисов

Для поддержания непрерывной работы сервисов на .NET Core необходимо создать службу systemd. Для этого необходимо создать файл с описанием службы - например, **/etc/systemd/system/capcon-reports.service**

Файл службы имеет следующее содержимое:

```
[Unit]
Description=Сервис отчетов

[Service]
WorkingDirectory=/home/webuser/dotnet/capcon-reports
ExecStart=/usr/bin/dotnet capcon-reports.dll
Restart=on-failure
RestartSec=20
SyslogIdentifier=dotnet-aisks-report
User=webuser
Environment=ASPNETCORE_ENVIRONMENT=Production

[Install]
WantedBy=multi-user.target
```

Служба запускается командой:

sudo systemctl start capcon-reports.service

Для обеспечения ее непрерывной работы выполняется команда:

sudo systemctl enable capcon-reports.service

Это позволит службе автоматически перезапускаться при перезапуске сервера и других возможных сбоях.

При любых изменениях сервиса необходимо перезапустить службу командой:

sudo systemctl restart capcon-reports

Для проверки состояния службы можно выполнить следующие команды:
sudo systemctl status capcon-reports
sudo journalctl -u capcon-reports

3.3.3 Перечень сервисов
3.3.3.1 Интеграция с 1С/МДМ

- Расположение: /home/webuser/dotnet/mdm.integration/
- Systemd: /etc/systemd/system/mdm.integration.service

3.3.3.2 Интеграция с МТП

- Расположение: /home/webuser/dotnet/ext.mtp.integration/
- systemd: /etc/systemd/system/ext.mtp.integration.service
- работает с очередью МТП RabbitMQ на хосте **mq-rabbit-003**

3.3.3.3 Интеграция с Парус

- Вебсервис:
  - Расположение: /home/webuser/dotnet/ext.parus.integration/
  - Systemd: /etc/systemd/system/ext.parus.integration.service
  - Порт: **5003**
  - Адрес: http://capspro.infoenergo.loc:5003/ParusIntegrationService.svc
- Отправка в Парус по crontable:
  - Расположение: /home/webuser/dotnet/ext.parus.integration.client/
  - Запускалка: /home/webuser/dotnet/start_parus_client.sh
  - Логи: /var/log/cron, /home/webuser/dotnet/ext.parus.integration.client/log/
  - Использует вебсервисы Паруса по адресам:
    - http://10.3.8.137/KSConEx/KSConExService.asmx
    - http://10.3.8.137/KSContracts/KSContractsService.asmx
    - http://10.3.8.137/KSContracts/KSSendingService.asmx
  - Настройка crontable: ***/1 * * * * /home/webuser/dotnet/start_parus_client.sh** (1 раз в минуту)

3.3.3.4 Интеграция с СБИС

- Расположение: /home/aisks/dotnet/ext.sbis.integration/
- Systemd: /etc/systemd/system/ext.sbis.integration.service

3.3.3.5. Сервис интеграции с Инвестпрограммой

- Расположение: /home/webuser/dotnet/invest.integration/
- Systemd: /etc/systemd/system/invest.integration.service

3.3.3.6. Сервис печати отчетов

- Расположение: /home/webuser/dotnet/capcon-reports/
- Systemd: /etc/systemd/system/aisks-reports.service

3.3.3.7. Сервис извлечения комментариев из документов:

- Расположение: /home/webuser/dotnet/extractor
- Systemd: /etc/systemd/system/ doc-extractor.service

3.4 Обеспечение доступа к серверам АИС КапС.Pro по протоколу HTTPS

3.4.1 Установка и обновление SSL-сертификатов для внутренних серверов АИС КапС.Pro

Для генерации SSL-сертификата в удостоверяющий центр Заказчика направляется соответствующий CSR-запрос. Файл запроса с расширением .csr генерируется на соответствующем сервере (на который необходимо установить сертификат) при помощи следующей команды:

openssl req -out aisks.csr -newkey rsa:2048 -nodes -keyout aisks.key -config /path/to/san.conf

где /path/to/san.conf - путь до файла конфигурации san.conf со следующим содержимым:

```
[ req ]
default_bits       = 2048
distinguished_name = req_distinguished_name
req_extensions     = req_ext
[ req_distinguished_name ]
countryName                = Country Name (2 letter code)
stateOrProvinceName        = State or Province Name (full name)
localityName               = Locality Name (eg, city)
organizationName           = Organization Name (eg, company)
commonName                 = Common Name (e.g. server FQDN or YOUR name)
[ req_ext ]
subjectAltName = @alt_names
[alt_names]
DNS.1   = capspro.infoenergo.loc
DNS.2   = app-capspro-163.infoenergo.loc
IP.1    = 10.3.78.4
```

Параметры *DNS.<n>*, *IP.<n>* указываются в соответствии с настройками сервера, для которого генерируется сертификат. В данном примере указаны настройки внутреннего сервера приложений АИС КапС.Pro app-capspro-163 (10.3.78.4).

В результате выполнения команды будут сформированы файлы закрытого ключа aisks.key и файл запроса aisks.csr, который необходим для генерации SSL-сертификата. Сгенерированный файл сертификата необходимо разместить на соответствующем сервере АИС КапС.Pro.

Файлы сертификатов и закрытых ключей располагаются в директориях /home/webuser/ssl основного и тестового серверов АИС КапС.Pro.

По истечении срока действия сертификатов, новые запросы могут быть сгенерированы той же командой. Они должны быть переданы сотрудникам информационной безопасности для генерации новых SSL-сертификатов в удостоверяющем центре Заказчика.

3.4.2 Обеспечение доступа по протоколу HTTPS для сервера Личного Кабинета АИС КапС.Pro

Доступ к серверам Личного Кабинета АИС КапС.Pro по протоколу HTTPS обеспечивается при помощи прокси-сервера (HAProxy). Поддержка прокси-сервера осуществляется сотрудниками Заказчика.

Запросы к основному веб-сервису Личного Кабинета (https://capspro-lk.infoenergo.ru) перенаправляются на порт 80 сервера Личного Кабинета app-capspro-162 (192.168.12.5). Запросы к тестовому веб-сервису Личного Кабинета (https://capspro-lk-test.infoenergo.ru) перенаправляются на порт 8080 сервера Личного Кабинета.

4. Диагностика неисправностей системы

Для централизованного сбора логов используется корпоративный сервер Graylog.
Для мониторинга состояния сервисов используется корпоративное ПО Zabbix. Управление данными сервисами осуществляет персонал Заказчика

4.1. Диагностика сервисов под управлением Node.JS
4.1.1. Для запуска монитора состояния сервисов под управлением Node.JS можно выполнить команду:
pm2 monit
4.1.2. Для просмотра логов выполняется команда:
pm2 logs <название или id сервиса> --lines <количество последних строк лога>
4.2. Диагностика состояния СУБД
Логи базы данных хранятся в течение 7 дней и располагаются в директории /opt/Postgresql/data/log на сервере БД.
4.3. Диагностика сервисов под управлением .NET Core
Для проверки состояния службы можно выполнить следующие команды:
sudo systemctl status <название сервиса>
sudo journalctl -u <название сервиса>